

[19] 中华人民共和国国家知识产权局



[12] 发明专利说明书

专利号 ZL 200480010027.3

[51] Int. Cl.

G06T 3/00 (2006.01)

G06T 15/20 (2006.01)

G06T 15/50 (2006.01)

[45] 授权公告日 2007 年 12 月 26 日

[11] 授权公告号 CN 100357973C

[22] 申请日 2004.4.13

[21] 申请号 200480010027.3

[30] 优先权

[32] 2003.4.15 [33] EP [31] 03076102.7

[86] 国际申请 PCT/IB2004/050422 2004.4.13

[87] 国际公布 WO2004/093010 英 2004.10.28

[85] 进入国家阶段日期 2005.10.14

[73] 专利权人 皇家飞利浦电子股份有限公司

地址 荷兰艾恩德霍芬

[72] 发明人 巴特·G·B·巴伦布吕格

科内利斯·迈因德斯

[56] 参考文献

WO0004505A1 2000.1.27

US6297833B1 2001.10.2

"3 - D TRANSFORMATIONS OF IMAGES IN SCANLINE ORDER". ED CATMULL, ET AL.: ACM. 1980

"Resample Hardware for 3D Graphics". KOEN MEINDS, ET AL.: GRAPHICS HARDWARE,. 2002

"Separable image warping with spatial lookup tables". GEORGE WOLBERG, ET AL.: COMPUTER GRAPHICS, Vol. 23 No. 3. 1989

审查员 许 馨

[74] 专利代理机构 中科专利商标代理有限责任公司

代理人 王波波

权利要求书 2 页 说明书 17 页 附图 5 页

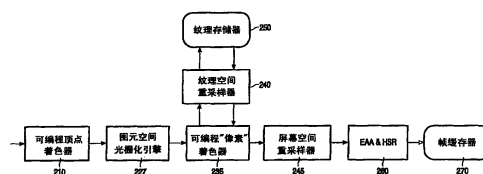
[54] 发明名称

计算机图形处理器及生成计算机图形图像的方法

[57] 摘要

描述了一种计算机图形处理器，包括：模型信息提供单元(210)，其提供表现一组图形基元的信息；光栅化引擎(227)，其能够生成第一序列坐标，该第一序列坐标与关联于图元的基本栅格一致；色彩生成器(235)，其为所述第一序列坐标分配色彩；以及显示空间重采样器(245)，其在第一和第二变换中，对色彩生成器在基本栅格内为坐标 u、v 分配的色彩进行重采样，重采样为与利用坐标为 x、y 的显示关联的栅格内的表示。该变换通过第一和第二遍执行，并有选择地包括转置。这两遍的次序以及决定是否进行转置取决于偏导数公式(1)的计算，其中的两个偏导数决定切变，另两个偏导数决定变换中的缩放。所作出的选择中，相对较大的偏导数作为缩放因子出现，相对较小的偏导数

作为切变因子出现，并且其中，最低的切变量出现在第二次变换中。



1、一个计算机图形处理器，包括：模型信息提供单元，用于提供表示一组图形基元的信息；光栅化引擎，其能够生成第一序列坐标，所述第一序列坐标与关联于所述图形基元的基本栅格一致；色彩生成器，用于为所述第一序列坐标分配色彩；以及显示空间重采样器，用于在第一和第二一维变换中，对所述色彩生成器在所述基本栅格内为坐标 u 、 v 分配的色彩进行重采样，以作为与利用坐标 x 、 y 的显示关联的栅格内的表示，所述第一和第二一维变换在第一和第二遍滤波中执行，并有选择地包括转置，

所述处理器有一个选择机构，用于选择所述一维变换的次序以及基于以下偏导数的计算来选择是否进行转置，

$$\frac{\partial x}{\partial u}, \frac{\partial x}{\partial v}, \frac{\partial y}{\partial u}, \frac{\partial y}{\partial v}$$

其中的两个偏导数决定切变，另两个偏导数决定所述一维变换中的缩放，所作出的选择中，相对较大的偏导数作为缩放因子出现，相对较小的偏导数作为切变因子出现，并且其中，最低的切变量出现在第二次一维变换中。

2、如权利要求 1 所述计算机图形处理器，其中，所述光栅化引擎在一个坐标系内的栅格上光栅化所述图形基元，所述坐标系提供了所述图形基元表面的二维参数化。

3、如权利要求 2 所述计算机图形处理器，其中，所述栅格是从纹理贴图中的一个得来的。

4、如权利要求 3 所述计算机图形处理器，其中，如果有一个以上纹理符合条件，则选择在纹理空间内具有最大区域的纹理。

5、如权利要求 2 所述计算机图形处理器，其中，如果没有与所述图形基元关联的纹理，则通过为每个顶点分配虚构纹理坐标，来选择所述图形基元表面上的虚构栅格。

6、生成计算机图形图像的方法，包括以下步骤：

- 提供表示一组图形基元的信息，

- 生成第一序列坐标，所述第一序列坐标与关联于所述图形基元的基本栅格一致，

- 利用表示所述图形基元的信息，来为所述第一序列坐标分配色彩，

- 在第一和第二遍滤波的第一和第二一维变换中，对在基本栅格内为坐标 u 、 v 分配的色彩进行重采样，以作为与利用坐标 x 、 y 的显示关联的栅格内的表示，

计算所述利用坐标 x 、 y 的显示相对于所述基本栅格中的坐标的偏导数 $\frac{\partial x}{\partial u}, \frac{\partial x}{\partial v}, \frac{\partial y}{\partial u}, \frac{\partial y}{\partial v}$ ，其中的两个偏导数决定切变，另两个偏导数决

定一维变换中的缩放，在所作出的选择中，相对较大的偏导数作为缩放因子出现，相对较小的偏导数作为切变因子出现，并且其中，最低的切变量出现在第二一维变换中。

计算机图形处理器及生成计算机图形图像的方法

技术领域

本发明涉及一种计算机图形处理器。

本发明还涉及一种生成计算机图形图像的方法。

背景技术

为多个不同目的,通过采用多种不同技术实现计算机生成图像的呈现。大量消费级产品都基于屏幕空间扫描线算法,该算法通过遍历屏幕上的像素,来逐个呈现投影到该屏幕上的图元。反向纹理映射(inverse texture mapping)用以决定图元对这样一个像素所贡献的色彩。通过该图形流水线后面几个阶段(光栅化及其后)的修改[8],前向纹理映射(forward texture mapping)也得以使用。通过前向纹理映射,遍历(traverse)图元在纹理贴图上的投影,而不是屏幕上的投影,并将扫描中所碰到的纹理采样(纹理像素)映射和喷涂(splatted)到屏幕上。该映射包括了一个二维(2D)变换。

已知,二维滤波的大部分滤波器可被近似看作使用两个一维(1D)滤波器,从而提供更高效的解决方法。见[2、17]中例子。

如下所述,这样的分离会带来所谓的瓶颈(bottleneck)和切变(shear)问题,且分别导致模糊(blurriness)和锯齿(aliasing)。为了缓和这些不利效应带来的后果,这些公开描述了四种遍历次序。从这四种遍历次序中选择一种来减轻瓶颈问题。超级采样(supersample)作为对切变锯齿的一种解决方法,在已知的文献中得到描述。超级采样需要一个额外的下变换(downscale)滤波器,以降低额外的分辨率从而达到要求的输出分辨率。

发明内容

此发明的目的是提供一种计算机图形处理器以及产生计算机图形图像的方法,其中,上述额外的下变换滤波器是多余的。本发明中,

与该目的相一致的计算机图形处理器将被描述。本发明中，与该目的相一致的生成计算机图形图像的方法将被描述。在根据本发明的计算机图形处理器中，模型信息提供单元提供了代表一组图形基元的信息，这些图形基元诸如三角形、其他多边形或贝塞尔形状。所提供的信息可以包括定义了图元形状的几何信息，和 / 或定义了图元外观的外观信息，诸如纹理和色彩信息。

根据本发明，提出了一个计算机图形处理器，包括：模型信息提供单元，用于提供表示一组图形基元的信息；光栅化引擎，其能够生成第一序列坐标，所述第一序列坐标与关联于所述图形基元的基本栅格一致；色彩生成器，用于为所述第一序列坐标分配色彩；以及显示空间重采样器，用于在第一和第二一维变换中，对所述色彩生成器在所述基本栅格内为坐标 u 、 v 分配的色彩进行重采样，以作为与利用坐标 x 、 y 的显示关联的栅格内的表示，所述第一和第二一维变换在第一和第二遍滤波中执行，并有选择地包括转置，

所述处理器有一个选择机构，用于选择所述一维变换的次序以及基于以下偏导数的计算来选择是否进行转置，

$$\frac{\partial x}{\partial u}, \frac{\partial x}{\partial v}, \frac{\partial y}{\partial u}, \frac{\partial y}{\partial v}$$

其中的两个偏导数决定切变，另两个偏导数决定所述一维变换中的缩放，所作出的选择中，相对较大的偏导数作为缩放因子出现，相对较小的偏导数作为切变因子出现，并且其中，最低的切变量出现在第二次一维变换中。

光栅化引擎能够生成第一序列坐标，该第一序列坐标与关联于图元的基本栅格 u 、 v 相一致，该光栅化引擎还能够进一步生成与该第一序列关联的一个或多个序列内插值。该进一步生成的序列可以包括屏幕(显示)坐标 x 、 y ，以及例如包括图元所表现出的表面的一般信息。

设置色彩生成器利用所述外观信息为所述第一序列坐标分配色彩。该色彩生成器可以简单地使用由光栅化引擎提供的内插色彩，但也可以进行复杂的着色和纹理操作。设置显示空间重采样器对色彩生成器在基本栅格中为坐标 u 、 v 所分配的色彩进行重采样，重采样为

与利用坐标 x 、 y 的显示关联的栅格中的表示。该变换通过第一和第二变换执行，其大大降低了计算量。

显示空间重采样器利用转置机构和选择机构，根据图像数据是否被转置，以及根据执行第一和第二变换的次序，即，首先计算显示坐标的 x 坐标还是 y 坐标，来从四个选项中进行选择。根据本发明的计算机图形处理器和方法包括了一种从四个选项中进行选择的新选择准则，其还把切变考虑在内。发明者发现，第二遍滤波器可以作为在第一遍中进行的任何超级采样的下变换滤波器，而在第二遍的超级采样则需要一个额外的下采样滤波器(第三遍)。因此，第一选择阶段为了减小瓶颈，其中决定了是否应用转置，该第一步后，基于这两遍(变换)中所出现的切变在剩余的两个选项中进行选择。

转置意味着包括了变换，其中，坐标对中的坐标互换。举例来说，可通过相对于直线 $x=y$ 或 $x=-y$ 对坐标作镜像来完成。通过前向或后向旋转 90° 也可以达到相同效果。但是，在这种情况下，如果第一转置是后向旋转，那么第二次转置应是前向旋转，反之亦然。

被选中的选项应使最低的切变量出现在第二遍中，从而避免在第

二遍中需要超级采样。或者，最坏的切变被放到第一遍中，其中可以很容易地利用超级采样来减少切变锯齿。

具体地，四个选项可以被进一步定义如下：

第一选项，其中：

在第一遍中，对于具有坐标 v 的每条线，通过将 u 映射到 x ， uv 图像数据被重采样为 xv 图像数据，以及

在第二遍中，对于具有坐标 x 的每条线，通过将 v 映射到 y ， xv 图像数据被变换为 xy 图像数据，

第二个选项，其中：

在第一遍中，对于具有坐标 u 的每条线，通过将 v 映射到 y ， uv 图像数据被重采样为 uy 图像数据，以及

在第二遍中，对于具有坐标 y 的每条线，通过将 u 映射到 x ， uy 图像数据被变换为 xy 图像数据，

第三个选项，其中：

在第一遍中，对于具有坐标 v 的每条线，通过将 u 映射到 y ， uv 图像数据被变换为 yv 图像数据，

在第二遍中，对于具有坐标 y 的每条线，通过将 v 映射到 x ， yv 图像数据被变换为 yx 图像数据，以及

yx 图像数据被转置为 xy 图像数据，

第四个选项，其中：

在第一遍中，对于具有坐标 u 的每条线，通过将 v 映射到 x ， uv 图像数据被变换为 ux 图像数据，

在第二遍中，对于具有坐标 x 的每条线，通过将 u 映射到 y ， ux 图像数据被变换为 yx 图像数据，以及

yx 图像数据被转置为 xy 图像数据。

在第三和第四选项中，对光栅化引擎所产生的坐标进行直接转置是不必要的。取而代之的是，可在生成基本栅格的坐标之前，将用于定义图元的顶点和/或控制点的坐标进行简单的转置。

附图说明

参考附图详细描述本发明的这些及其他方面，其中：

图 1 示出了计算机图形处理器的一种已知结构；

图 2 示出了适于实现本发明的计算机图形处理器的第一结构；

图 3 说明了对于两步图像变换所遇到的第一个问题；

图 4 说明了对于两步图像变换所遇到的第二个问题；

图 5 示出了适于实现本发明的计算机图形处理器的第二结构；

图 6A 示出了利用图 1 所示的已知计算机图形处理器所呈现出的图像；

图 6B 示出了利用图 5 所示的根据本发明的计算机图形处理器所呈现出的图像。

在这些图中，彼此对应的部分所具有的参考号码相差 100 或 100 的倍数。

发明详述

图 1 是一个图形流水线中最后几个阶段的结构，其特点是可编程顶点和像素着色。可编程顶点着色器 10(与位于流水线末端的帧缓冲器 70 一起)向参考图 2 等图所述的结构提供了变化的上下文。对于所呈现图元的每个顶点，顶点着色器[6]可为像素着色器 30 修改和建立数据。顶点着色器 10 向光栅化引擎 20 提供的数据(用于内插)通常包括一些属性，如漫反射色彩(diffuse color)和/或高光反射色彩(specular color)、纹理坐标、(齐次)屏幕坐标、以及有时出现的如曲面法线这样的额外数据，或着色处理需要的其他数据。

将这些属性提供给屏幕空间光栅化引擎 20，该光栅化引擎 20 通过从这些顶点属性中选择屏幕坐标作为光栅化处理的驱动变量，使用扫描线算法来遍历位于图元在屏幕上的投影内的像素。光栅化引擎 20 内插这些属性，从而为给每个被访问的像素赋值。该内插解决了从世界空间到屏幕空间的透视映射。因而这些属性在每个像素位置上都是可用的，其中像素着色器[11、12、13]可利用它们来计算局部表面色彩(local surface color)。当需要纹理数据时，利用纹理空间重采样器 40，基于纹理坐标从纹理存储器 50 中获得纹理采样。基于从光栅

化引擎 20 接收的内插坐标, 以及基于来自先前纹理提取的(所谓的相关纹理(dependent texturing))和/或经计算得到的任何结果, 由像素着色器 30 生成这些纹理坐标。纹理滤波操作通常基于临近纹理像素的双线性或三线性内插, 或基于这样的纹理探测(texture probes)的结合, 来近似一个各向异性的(透视变换)滤波器覆盖范围(footprint)。

当一个像素的表面颜色被确定后, 所得到的像素片断被继续发送到边缘抗锯齿失真和隐藏表面消除(EAA & HSR)单元 60。通常, 该单元针对隐藏表面消除使用 Z 缓存, 并针对边缘抗锯齿失真使用多重采样(利用相关的子采样缓存和下采样逻辑)。当所有的图元都被呈现时, 使用预滤波器(通常为 box 滤波器)进行下采样, 从而将子采样合成为最终像素分辨率的色彩。

当呈现高保真图像时, 一个重要的问题就是抗锯齿失真(包括纹理抗锯齿失真和边缘抗锯齿失真)。当前的超级采样和多重采样技术的计算强度高, 需要大量的存储带宽。在使用前向纹理映射的图形流水线结构中, 通过容易地使用预滤波可避免锯齿失真[8、9、2]。这样的前向纹理映射结构如图 2 所述。

如图 1 的传统流水线, 图 2 中所示的计算机图形处理器包括模型信息提供单元 110, 该单元也被表示为可编程顶点着色器, 或被表示为转换与光照单元(transformation and lighting unit), 以提供代表一组图形基元的信息。该计算机图形处理器还包括光栅化引擎 125, 其能够生成第一序列坐标, 这些坐标与关联于图元的基本栅格(grid)相一致。

与传统屏幕空间光栅化引擎 20 不同, 纹理空间光栅化引擎 125 通过选择纹理坐标(而不是屏幕坐标)作为光栅化处理的驱动变量, 遍历图元在纹理贴图(而不是屏幕)上的投影。所有属性都被内插(线性内插, 除了纹理像素被投影到其上的屏幕坐标之外, 该屏幕坐标被透视内插)。例如在对应于四维(4D)MIP 映射(mipmapping, 一种多纹理映射技术)(关于四维 MIP 映射的更多细节, 见 Heckbert 的关于纹理映射的概述[4])的栅格上, 纹理空间光栅化引擎 125 遍历纹理贴图。然后, 纹理拾取从存储在纹理存储器 150 中的三维(3D)MIP 映射数据得到 4

维 MIP 映射重构(mipmap reconstruction)。或者, 纹理可被无压缩地存储在存储器 150 中, 或者可以以飞速写入方式(on the fly)生成。

计算机图形处理器还包括了色彩生成器(未示出), 来为上述的第一序列坐标分配色彩。其中, 被拾取的纹理像素可由内插的满反射和/或高光反射色彩来合成, 从而产生一个表面的色彩采样, 其具有相关联的屏幕坐标(通常非整数), 该屏幕坐标表示这个纹理采样映射到屏幕上的位置。

色彩生成器在基本栅格内为坐标 u 、 v 所分配的色彩, 被显示空间重采样器 145 重采样为在与利用坐标 x 、 y 的显示关联的栅格中的表示。

然后, 来自于屏幕空间重采样器 145 的像素片断(pixel fragments)在 EAA & HSR 单元 160 中被合成, 该单元使用了一个片断缓存器[8]。像素片断被按深度分类(depth-sorted), 保存到该缓存器内, 以解决隐藏表面问题。在所有的图元都被呈现后, 每个像素的所有可视片断被合成在一起(其可以相当于简单求和, 因为屏幕空间重采样器 145 传递的色彩已被预滤波器加权), 并被发送到帧缓存器 170。边缘抗锯齿失真由屏幕空间重采样器 145 在临近边缘处产生的部分贡献的结合引起的, 其最终导致由来自不同图元的色彩所合成的像素色彩。在 A-缓存[1]、Z3 缓存[5]或三维引擎(WarpEngine)[14]中使用的技术, 也可用于 EAA & HSR 单元中, 尽管 Meinds 的文章[8]中所述的片断缓存器与这些技术不同之处在于, 缓存器中来自于一个像素的色彩被看作对该像素的整个滤波器覆盖范围(可能彼此会部分重叠)的部分贡献, 而不是超级采样栅格(总是一个接着一个)上的某一位置的色彩。

由像素着色处理得到的着色的色彩采样连同其屏幕坐标一起被发送到屏幕空间重采样器 145。屏幕空间重采样器 145 将色彩样本(一般位于非整数像素位置)重采样到显示所需的整数像素位置。即, 屏幕空间重采样器将映射的纹理像素喷涂到整数屏幕位置上, 并在屏幕上提供图元的图像。二维重采样操作可以通过第一和第二一维变换被高效地执行[2], 例如, 使用一维 FIR 滤波器结构, 并有选择地包括转置(transposition)。

在由相同发明者于 2003 年 2 月 13 日申请的欧洲专利申请 03100313.0 中描述了光栅化引擎和色彩生成器,其为模型信息提供单元的可能实现方式。

根据本发明,处理器具有一个选择机构,用于对变换的次序进行选择,并基于偏导数 $\frac{\partial x}{\partial u}, \frac{\partial x}{\partial v}, \frac{\partial y}{\partial u}, \frac{\partial y}{\partial v}$ 的计算来选择是否使用转置,这四个偏导数中,两个决定变换中的切变,两个决定变换中的缩放(scaling)。选择出的相对较大的偏导数作为缩放因子出现,相对较小的偏导数作为切变因子出现,其中,在第二变换中出现最低切变量。

该选择过程可进一步阐明如下。

如果屏幕空间重采样器 245 通过两遍滤波(2-pass filtering)来实现,光栅化引擎 227 除了为图元表面遍历做准备,还必须为两遍滤波做准备。已知的是,两遍滤波会受到瓶颈和切变问题的影响[17、15、2]。

瓶颈问题如图 3 所示,其中,中间图像 II 的区域相对于输入和输出图像 TI、SI 变得很小。其以接近 90 度的旋转出现,并在第二遍方向上导致额外的模糊(因为第二遍必须再次将折叠后的中间图像放大)。

切变问题如图 4 所示,其中,示出了纹理贴图 TI 中的两行以及中间图像 II。纹理贴图 TI 有一条黑色的垂直线,并且透视映射的切变是这样的,第二行中的黑像素位置相对第一行中的黑像素位置向右移了五个像素。一遍水平滤波只能去除水平锯齿,但不能去除垂直锯齿。切变导致了在一行中黑像素和在下一行中白像素之间很明显的跃迁。而且,中间图像 II 中的这条线由不相连的部分组成,该不相连的部分被该线对其没有贡献的列(例如, $x=3$ 处)所分离。对于切变问题的一种解决方法[17]是以更高的分辨率来光栅化,这样会导致产生额外的中间线,使得在多个空白的中间位置填入黑像素。如果对第一遍的切变执行该操作,那么第二遍将会对较大的中间图像进行下采

样,以达到其最终的分辨率。如果使用相同的超级采样方法来处理第二遍中的切变,则需要有第三遍来降低所生成的较高水平分辨率,以达到输出分辨率。

与 Catmull 和 Smith 的文章[2]十分相似,本发明通过在四个选项中进行选择来避免瓶颈问题,这个选择通过决定以下两点来获得,(1)决定立刻产生输出图像 SI,或产生转置的版本并转置所生成的图像,以及,(2)决定首先进行一遍水平处理,还是首先进行一遍垂直处理。但是,根据本发明,使用不同的标准从这些选项中进行选择,以最小化第二遍的切变。通过这种方法,不必生成额外的图像列来处理第二遍中的切变。在最坏情况下,只需产生中间的线来处理第一遍中的切变锯齿。这就避免了进行第三遍的代价。考虑到在某一点 p 周围映射的局部线性:

$$\begin{pmatrix} x - x_p \\ y - y_p \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{pmatrix} \begin{pmatrix} u - u_p \\ v - v_p \end{pmatrix}$$

这里所使用的偏导数是瓶颈和切变问题(在点 p 处)的很好的指示器:第一遍中 u 映射到 x 且在第二遍中 v 映射到 y 的情况下, $\frac{\partial x}{\partial u}$ 是第一遍的缩放因子:如果该值小(接近于零),则中间图像将会折叠,并表现出瓶颈问题。 $\frac{\partial y}{\partial v}$ 是第二遍的缩放因子,以及 $\frac{\partial x}{\partial v}$ 和 $\frac{\partial y}{\partial u}$ 分别是第一遍和第二遍的切变量。对于处理瓶颈问题的四个选项之一,四个偏导数中的每一个都起到第一遍缩放因子的作用。选择对应于最大的第一遍缩放因子的选项(从而最大化中间图像区域[2]),减少了瓶颈问题,但是会把切变问题留给第二遍,而不是第一遍。根据本发明,通过两个阶段在四个选项中进行选择,考虑到了切变问题。

第一,决定如何配对坐标:将 u 映射到 x ,将 v 映射到 y ,或者将 u 映射到 y ,将 v 映射到 x 。基于图元顶点处的偏导数以及一方面

u 和 v 之间的相关量和另一方面 x 和 y 之间的相关量,确定上述映射。通过选择相关性最强的两个偏导数作为缩放因子,来配对坐标。另两个偏导数成为切变因子。这就相当于 Catmull 和 Smith 所述的“转置与否”的选择。

第二,选择两遍处理的次序。交换两遍处理的次序,就是把第一遍的缩放和切变因子转置到第二遍当中,反之亦然。该次序的选择使得第二遍中具有最少的切变量。该选择也决定了两个缩放因子中的哪个是第一遍的缩放因子,哪个是第二遍的缩放因子。就瓶颈问题而言,这可以产生一个次优的选择,但是由于第一准则选择强相关的坐标作为缩放因子,所以两个缩放因子通常都足够大以避免瓶颈问题。这样,该选择处理在避免瓶颈和第二遍切变问题之间提供了很好的折衷。

像素着色通常相当于使用一种或另一种方法,进行几个纹理的合成。这可以通过如图 1 所示的传统结构来容易地完成,这是因为纹理空间重采样器 40 将所有纹理都重采样到普通屏幕像素栅格上,其中,可以对每个像素的纹理采样进行合成。

在参考图 2 所描述的前向纹理映射结构中,只有在屏幕空间重采样器阶段 145 之后,纹理采样在普通屏幕像素栅格上才是可用的。而且,光栅化引擎一次仅可以遍历一个栅格,所以,当如前向映射结构中一样正在纹理栅格上发生光栅化并使能多纹理化(multi-texturing)时,光栅化引擎必须从很多纹理中选择一个纹理,同时在相关联的栅格上遍历图元。多纹理可以以多遍(multi-pass)方式处理,使得在将这些纹理重采样到屏幕像素栅格后,可以对其进行合成。但是,这样会使 EAA & HSR 单元 160 中的片断缓存器拥塞。这样也会阻止高级特性,例如相关纹理处理,或具有相反性质的纹理模型,如环境映射凹凸贴图(environment mapped bump mapping)(其中,在每个栅格位置的凹凸映射信息决定了定位环境贴图的位置,可能会导致从环境贴图到图元表面和屏幕的一对多前向映射)。

如果屏幕空间重采样器可以在认为只有一个纹理贴图与图元关联的情况下映射纹理采样,那么这些问题就都可避免。因此,表面色彩的着色可以在屏幕空间重采样处理之前进行。其在图 5 中有相应描

述，该图 5 表示了一种混合图形流水线。

该混合流水线中，光栅化引擎在“表面栅格”上对图元进行光栅化，即，在为图元表面提供二维参数化的坐标系统中的栅格上进行光栅化。与纹理贴图关联的栅格提供这样一个表面栅格，并且优选地用作表面栅格(由于在纹理栅格上获取纹理采样不需要重采样)。但是，万一出现纹理贴图的空缺，或者例如纹理是一维或三维时，则可以选择另一栅格。其它段落中更详细地对其进行描述。除了(在前向纹理映射情况下)与每个栅格位置关联的透视映射屏幕坐标之外，可以在该栅格上线性地内插属性。

光栅化引擎 227 在该栅格上的多个位置处内插属性(包括二级纹理坐标)。然后，像素着色器 235 对该表面栅格的栅格位置上各个属性进行操作，并且如果有任何与图元关联的二级纹理(secondary texture)，其就使用标准纹理空间重采样器进行反向映射，以从这些纹理中获得色彩。如果该表面栅格从一个纹理贴图中被选择，那么该“初级(primary)”纹理阶段的纹理像素拾取从前向映射流水线浓缩为四维 MIP 映射重构处理。该处理是一种沿坐标轴的各向异性滤波的形式，且在标准的纹理空间重采样器中各向异性滤波通常是可用的。被拾取的纹理采样以及其他属性可被像素着色器程序使用，以计算当前栅格位置的表面色彩。

一旦表面栅格上的采样被着色，利用屏幕空间重采样器 245 将该色彩喷涂到屏幕像素上，其中，EAA & HSR 单元 260 可以将来自不同图元的贡献合成在一起。

使用图 5 所示的混合结构会产生若干结果，并具有若干吸引人的特性。对于混合流水线的不同阶段，来讨论这些结果和特性，并讨论这些阶段相对于图 1 所示的传统流水线的变化。

光栅化引擎 227 必须选择一个图元栅格，并避免瓶颈和切变问题，以及控制 MIP 映射。可编程的像素着色器 230 和纹理空间重采样器 240 保持与传统像素着色流水线中相同，且对于屏幕空间重采样器 245 有一些选择。

表面空间光栅化引擎 227 包含一个额外的第一流水线阶段(在正

常的建立和遍历阶段之后), 其中表面栅格被选定。该阶段在正常的光栅化引擎设置之前完成。优选地, 该表面栅格从纹理贴图中的一个得到, 从而不必通过纹理空间重采样器 240 重采样该纹理贴图(除可选的四维 MIP 映射之外)。因此, 栅格建立阶段可以检查与图元关联的纹理贴图。对于符合能够提供表面栅格这个条件的纹理贴图, 其必须满足三个要求。第一, 其必须不是相关寻址; 第二, 其必须是二维纹理(一维和三维纹理不适合遍历二维 2D 表面); 第三, 顶点的纹理坐标不应该组成退化的图元(例如, 其中所有的纹理坐标连成一线, 生成有效一维纹理)。

如果一个以上纹理符合条件, 则选取在纹理空间具有最大区域的纹理: 该纹理具有潜在的最多细节和最高频率(这样最能避免对于该纹理的纹理空间重采样处理, 因为该处理过程会导致不需要的模糊和锯齿)。

如果没有符合条件的纹理可用(例如, 假设有一个简单的高氏(Gouraud)着色的图元), 通过为每个顶点分配虚构的“纹理坐标”, 可以在图元的表面选择虚构的栅格, 使光栅化引擎 227 进行遍历。然后, 该虚构栅格如同纹理栅格一样(除对于这些坐标的纹理拾取被禁止外)被光栅化引擎遍历。正如 Chen[3]所提到的, 其优点是所得到的表面上的线性内插为透视提供了正确的高氏着色。将每个顶点的 x 和 y 屏幕坐标分配为虚构纹理坐标成为一种选择。注意, 该选择并不意味着光栅化引擎遍历屏幕像素, 因为当把一个“纹理像素”映射到屏幕上时, 每个顶点的齐次 w 仍被考虑在内。另一个选择(对于诸如三角形的平表面)是在眼睛空间内旋转三维顶点坐标, 使得被旋转表面的法线与眼睛空间 z 轴对准, 然后选择旋转的眼睛空间 x 和 y 坐标作为每个顶点的虚构栅格坐标。

在图元表面遍历任何栅格的能力提供了很大自由度。例如, 这种自由度可以被用来避免任何瓶颈和切变问题, 这些问题可能会与特定纹理栅格到屏幕的映射关联。这种自由度可以被用来遍历非平坦的图元(例如, 经由其表面参数化所决定的二维栅格遍历一个贝塞尔面片(Bezier patch), 并使用前向映射来直接将表面色彩喷涂到屏幕上)。光

栅化引擎也可以在图元的运动方向上利用这种自由度，从而可以沿着光栅化的方向应用一维滤波器，以高效地实现运动模糊[7]。

光栅化引擎 227 必须以某一分辨率对图元的表面进行采样，该分辨率对于其投射到屏幕上的图像 SI 的分辨率是合适的。因此，以与传统流水线中的 MIP 映射相似的方式对栅格进行遍历。当有或多或少的细节需求时，通过改变用作栅格坐标和内插属性的增量的 δ 值，光栅化引擎可以改变 MIP 映射级别(mipmap level)。这样，光栅化引擎在表面栅格上采取或大或小的步长。

光栅化引擎 227 经由透视映射来保持与每个栅格位置关联的屏幕坐标。利用这些坐标，通过确认后续栅格位置的屏幕坐标之间的差异保持在合适的范围之内(例如，在和 1 之间，但是，当然可以采用 MIP 映射级别偏差(mipmap level bias))，可以确定 MIP 映射开关(mipmap switch)是否状况良好。

上述处理并不直接提供用于从任何关联的纹理贴图拾取采样的 MIP 映射级别，这是由于它仅能保证一个合适的表面到屏幕的 MIP 映射级别。对于每个纹理贴图来说，还有一个与从纹理栅格到表面栅格的映射关联的缩放因子。该缩放因子对应一个纹理到表面的 MIP 映射级别，该纹理到表面的 MIP 映射级别可以被加到表面到屏幕的 MIP 映射级别上，以达到所需的总体的纹理到屏幕的 MIP 映射级别。

从用于在图元表面上内插纹理坐标的 δ 值可以直接获得纹理到表面的 MIP 映射级别，并且由于该内插是线性的，所以纹理到表面的 MIP 映射级别对每个图元来说是不变的，并且可以被光栅化引擎建立储存在与每个纹理重采样阶段关联的寄存器内，用作偏移量。

可编程“像素”着色器 230 和纹理空间重采样器 240(对于串行或并行拾取纹理采样，该采样器可以是一个或多个)与传统流水线中的对应部分完全相同。像素着色器 230 接收到某一位置的一组(内插的)属性，包括纹理和屏幕坐标。纹理坐标与着色程序一起经由纹理空间重采样器 240，来确定定位纹理贴图的位置。在把纹理坐标发送到纹理空间重采样器 240 以实现相关纹理之前，着色器 230 还可以改变纹理坐标，与传统的流水线中的方式完全相同。

可编程着色器 230 将着色的色彩以及通常为非整数的相关联的屏幕坐标一起传送给屏幕空间重采样器 245。可利用与传统像素着色器流水线(图 1)中像素着色器 30 类似的方法对其进行处理,当进行超级采样时该传统流水线可接收子像素(sub-pixel)屏幕位置。着色的色彩是对一个位置进行计算的结果,且并不依赖于被光栅化引擎 227 遍历的栅格。这就是说,现有的着色器程序无需进行修改,就可以在所推荐的结构上运行。

在表面空间进行可编程着色具有一些优点和缺点。除了前向映射所带来的高质量的抗锯齿失真之外,一个主要优点就是纹理空间重采样器 240 不必处理到屏幕的透视映射。

大部分时间里,可编程着色被用来进行从纹理栅格到另一栅格的仿射变换(affine transformation)。基于标准双线性/三线性探测的纹理空间重采样器,相比透视映射所需的更为一般的着色覆盖区域,能更好地近似该仿射映射所需的滤波器覆盖区域,因此这种重采样处理的质量将会比较高。仅有屏幕空间重采样器 245 必须处理透视重采样,且它仅会在表面栅格的着色采样上应用一次。

这种方法的一个缺点是,与传统的像素着色流水线相比,要对更多的采样进行着色。这是由于在每个方向上 MIP 映射保持了一个介于 1 和 2 之间的缩小因子(因此,对于每个像素,大约有 1.5x1.5 的表面元素被采样)。然而,高质量的抗锯齿失真将会确保子像素细节仍对最终的图像有所贡献,从而更进一步改善图像质量。

另一个缺点是现在二级纹理被两次重采样(一次被纹理空间重采样器 240,另一次被屏幕空间重采样器 245),这样可能会引入额外的模糊。通过选择具有最高细节的纹理贴图作为主纹理(primary texture),这就保证了最丰富的细节仅被重采样了一次。二级纹理将会有更小的缩小(或甚至会扩大,通常在如光照贴图的情况下出现),所以一些额外的模糊就不是很明显。屏幕空间重采样器 245 还使用了高质量锐度增强滤波器,其也能帮助保持清晰的图像。

图 6 表示了来自游戏 QuakeIII 中同一场景[16]的比较,该场景分别由图 1 的传统流水线以及图 5 混合流水线的样机所呈现。图 6a 展

示了例如在楼梯上的锯齿失真，即使采用了 2x2 的多采样，失真仍存在。在图 6b 中，虽然生成该图像所花费的计算量和带宽几乎和图 6a 的相同，而该锯齿失真几乎消失了。在地板上的影子和在墙上的灯光证明使用了光照贴图，其是利用混合流水线的样机中的多纹理“着色器”所呈现的。

参考文献

1. Loren Carpenter. The A-buffer, an antialiased hidden surface method (A-缓存，一种抗锯齿的隐藏表面方法). In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH'84 Proceedings)*, volume 18, pages 103-108, July 1984.
2. Edwin Catmull and Alvy Ray Smith. 3-d transformations of images in scan line order(扫描行顺序的图像三维变换). In *Computer Graphics (SIGGRAPH'80 Proceedings)*, volume 14(3), pages 279-285, 1980.
3. Baoquan Chen, Frank Dachille, and Arie Kaufman. Forward image mapping(前向图像映射). In *Proceedings of IEEE Visualization '99*, pages 89-96, October 1999.
4. Paul S. Heckbert. Survey of texture mapping(纹理映射概论). *IEEE Computer Graphics and Applications*, Nov.1986.
5. Norman P. Jouppi and Chun-Fa Chang. Z3: An economical hardware technique for high-quality antialiasing and transparency(用于高质量抗锯齿及透明的一种经济的硬件技术). In *Proceedings of Eurographics/Siggraph workshop on graphics hardware 1999*, pages 85-93. ACM Press, 1999.
6. Erick Lindholm, Mark J. Kilgard, and Henry Moreton. A user-programmable vertex engine(一种用户可编程的顶点引擎). In *Proceedings Siggraph 2001*, pages 149-158, August 2001.
7. Koen Meinds. Temporal anti-aliasing with forward texture mapping(使用前向纹理映射的时域抗锯齿失真). In *Submitted to Graphics Hardware 2003*, 2003.

8. Koen Meinds and Bart Barenbrug. Resample hardware for 3d graphics(三维图形的重采样硬件). In T. Ertl, W. Heidrich, and M Doggett, editors, *Proceedings of Graphics Hardware 2002*, pages 17-26, 2002.
9. The resample hardware for 3d graphics webpage at(三维图形的重采样硬件网页)
<http://www.extra.research.philips.com/graphics/>, 2003.
10. The Mesa homepage(Mesa 的主页) at <http://www.mesa3d.org/>
11. Microsoft. *DirectX Graphics Programmers Guide(DirectX 图形程序员指南)*. Microsoft Developers Network Library.
12. M.Olano, J.C.Hart, W. Heidrich, and M. McCool. *Real-time shadings(实时着色)*. A K Peters, Natick, Massachusetts, 2002.
13. M. Olano and A. Lastra. A shading language on graphics hardware: The pixelflow shading system(一种关于图形硬件的着色语言: 像素流着色系统). In *Proceedings Siggraph 1998*, pages 159-168, July 1998.
14. Voicu Popescu, John Eyles, Anselmo Lastra, Joshua Steinhorst, Nick England, and Lars Nyland. The warpengine: An architecture for the post-polygonal age(三维变形引擎: 一种用于多边形贴图时代的结构). In *Proceedings Siggraph 2000*, pages 433-442, 2000.
15. Alvy Ray Smith. Planar 2-pass texture mapping and warping(平面两遍的纹理映射和变形). In *Computer Graphics (Proceedings Siggraph 1987)*, volume 21(4), pages 263-272, 1987.
16. Quake III level "Subversive Tendencies" available at (Quake III 级别“颠覆趋向”可在如下网页找到)
<http://bettenberg.home.mindspring.com/teqtrny3.html> by Tequila, which uses textures maps from <http://www.planetquake.com/hfx/> Some of these are replaced, and for some the contrast is increased(使用了来自该网址的纹理贴图, 其中一些被替换, 一些的对比度提高).
17. George Wolberg and Terrance E. Boult. Separable image warping with spatial lookup tables(使用空间查找表的可分离的图像变形). In

Computer Graphics (*Proc. Siggraph '89*), volume 23(3), pages 369-378, July 1989.

18. Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. Surface splatting(表面喷涂). In *Proceedings Siggraph 2001*, pages 371-378, August 2001.

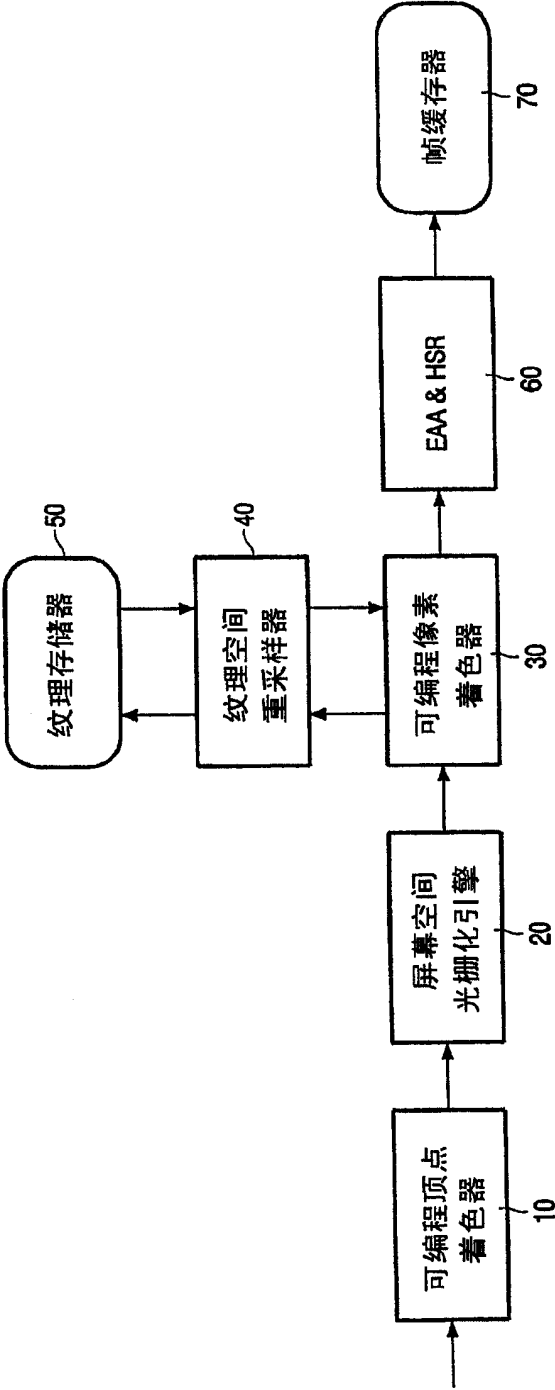


图1

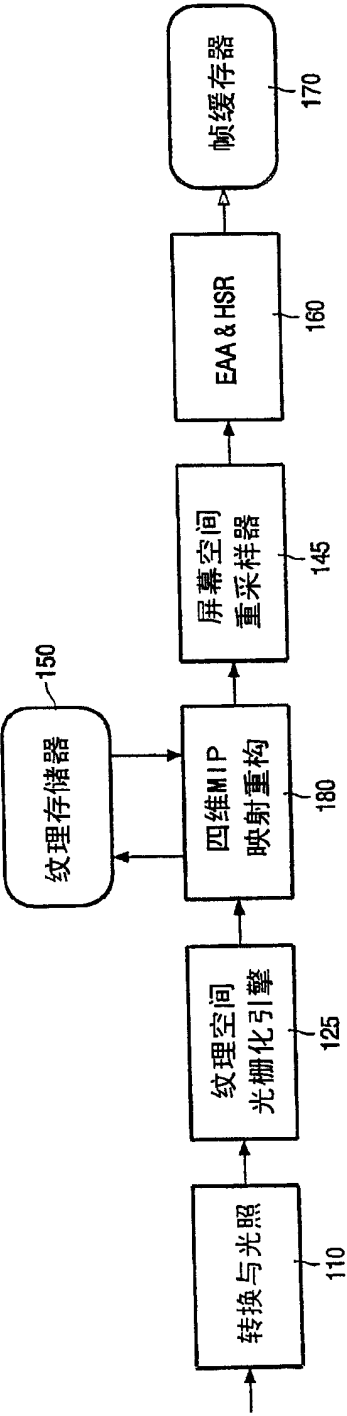


图2

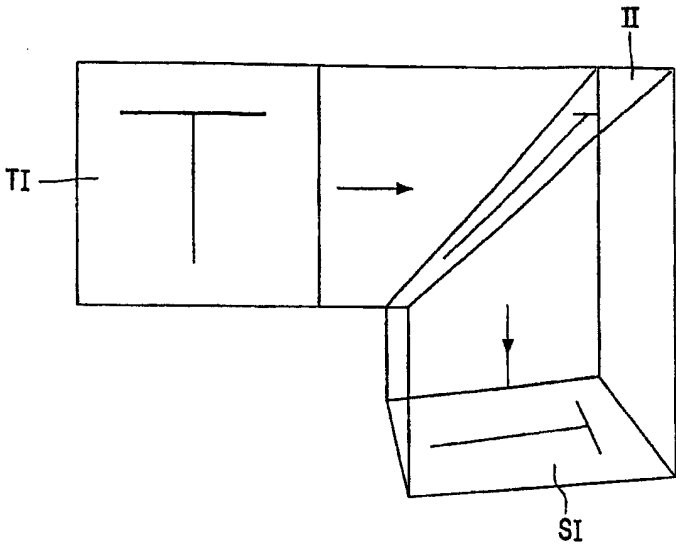


图3

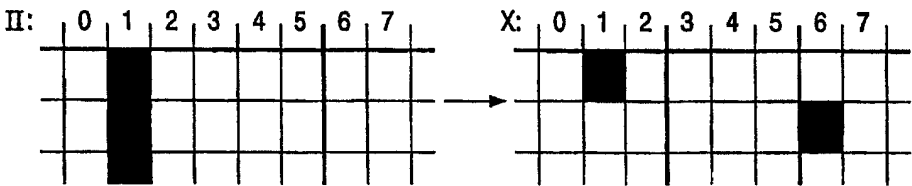


图4

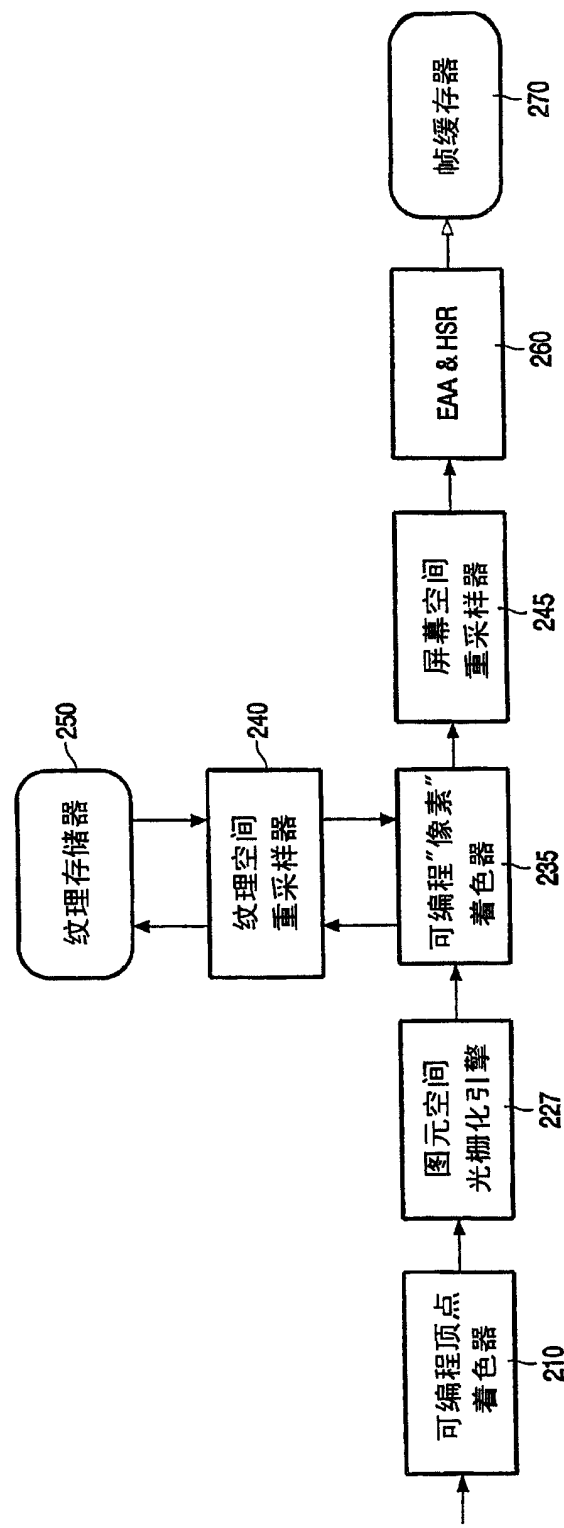


图5

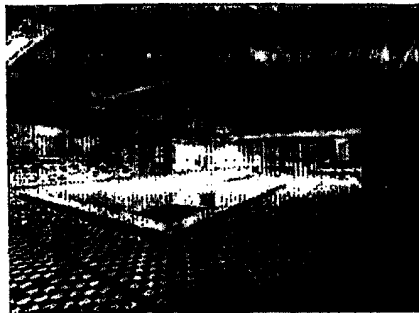


图6A



图6B